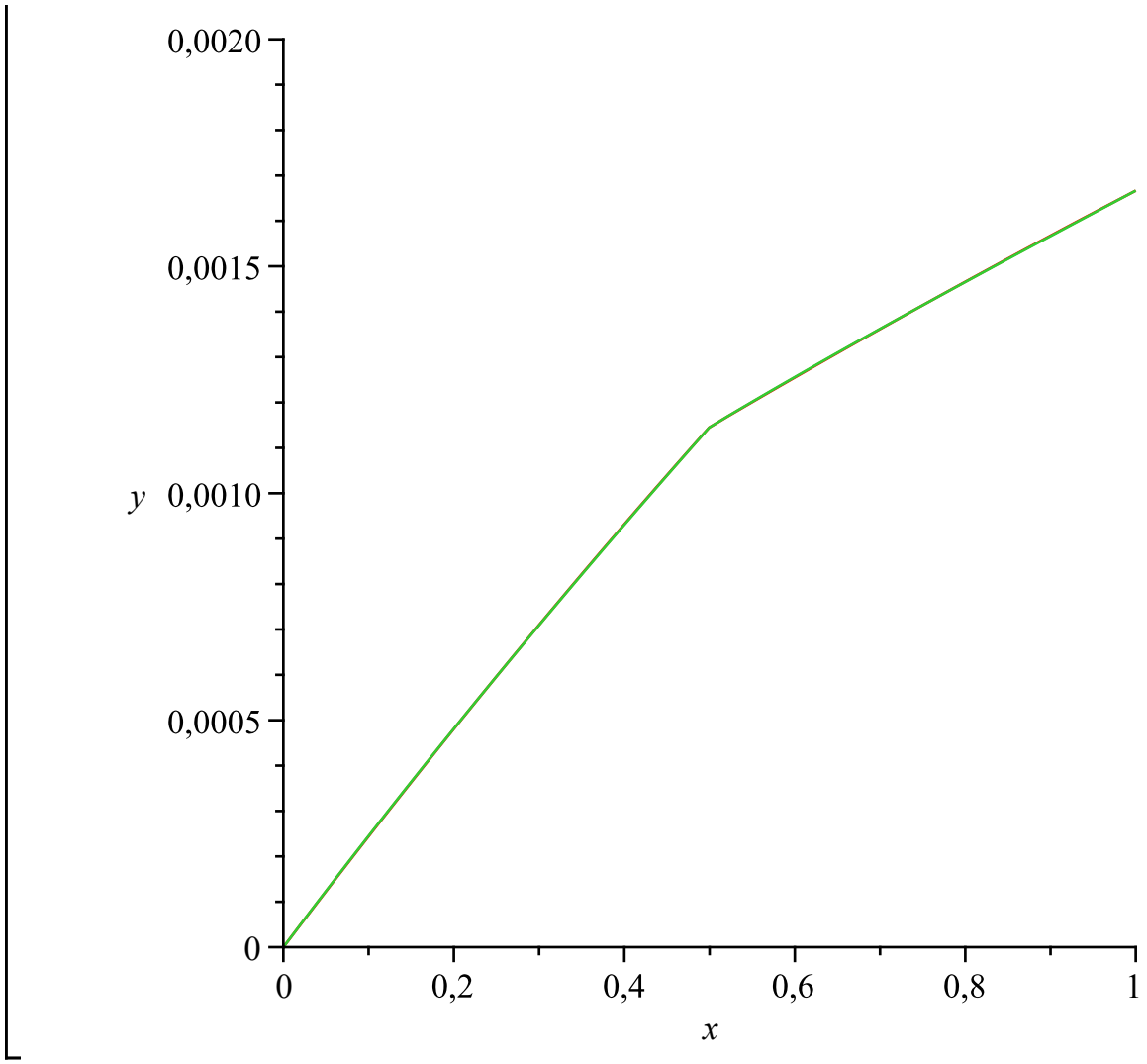
```
> with(LinearAlgebra) :
> phi[1] := xi -> piecewise(xi >= -1 and xi <= 0, (1 + xi)·(1 + 2·xi), xi > 0 and xi <= 1, (1
      - xi)·(1 - 2·xi)) :
> phi[2] := xi -> piecewise(xi >= -L/2 and xi <= L/2, (1 + 2·xi)·(1 - 2·xi)) :
> X := Matrix(5, 1, [0, L/4, L/2, 3·L/4, L]) :
> h := L/2 : L := 1 :
> Phi := Matrix(5, 1, [phi[1]((x - X[1, 1])/h), phi[2]((x - X[2, 1])/h),
      phi[1]((x - X[3, 1])/h), phi[2]((x - X[4, 1])/h), phi[1]((x - X[5, 1])/h)]) :
> plot([Phi[1, 1], Phi[2, 1], Phi[3, 1], Phi[4, 1], Phi[5, 1]], x = 0..1) :
> Rigidez := proc(m, EA); local i, j, M;
  description "obtém a matriz de rigidez";
  M := Matrix(m, m);
  for i from 1 to m do
  for j from 1 to m do
> M[i, j] := EA·int(diff(Phi[i + 1, 1], x)·diff(Phi[j + 1, 1], x), x = 0..L);
> end do;
> end do;
> return M;
> end proc:
> K := Matrix(4, 4) :
> K := Rigidez(4, 1e7) :
> Carga := proc(m, p0, F3, F5); local i, F;
> description "obtém o vetor de carga";
> F := Matrix(m, 1);
> for i from 1 to m do F[i, 1] := int(p0·(1 - x)·Phi[i + 1, 1], x = 0..1); end do;
> F[2, 1] := F[2, 1] + F3;
> F[4, 1] := F[4, 1] + F5;
> return F;
> end proc:
> f := Matrix(4, 1) :
> f := Carga(4, 1.0e4, 1.0e4, 1.0e4) :
> U := Multiply(MatrixInverse(K), f) :
> u := Phi[1, 1]·0 + Phi[2, 1]·U[1, 1] + Phi[3, 1]·U[2, 1] + Phi[4, 1]·U[3, 1] + Phi[5, 1]
      ·U[4, 1] :
> Uexata := piecewise(x >= 0 and x <= L/2, x/(6·L·EA)·(p0·(x^2 - 3·L·x + 3·L^2) + 12·F·L), x
      > L/2 and x <= L, 1/(6·L·EA)·(p0·x·(x^2 - 3·x·L + 3·L^2) + 3·F·L·(2·x + L))) :
> uexata := subs(p0 = 1.0e4, F = 1.0e4, L = 1, EA = 1.0e7, Uexata) :
> plot([u, uexata], x = 0..1, y = 0..0.002);
```


0,0020
0,0015
y
0,0010
0,0005
0
0
0,2
0,4
0,6
0,8
1
x